Claim Listing

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (withdrawn) A method of processing network data in a processor having multiple programmable multi-threaded engines integrated within the processor, the method comprising:

scheduling a first thread provided by any of the multiple programmable multi-threaded engines integrated within the processor to process a first incoming block of data within a network packet received at a port of a media access control device to move the first incoming block of data to a first location in a memory location in a memory coupled to the processor; and

scheduling a second thread provided by any of the multiple programmable multi-threaded engines integrated within the processor to process a second incoming block of data within the network packet to move the second incoming block of data to a second location in the memory prior to the first thread completing processing of the first incoming block of data.

2. (cancelled)

3. (withdrawn) The method of claim 1 further comprising:

saving state information by the first thread; and

retrieving the state information by the second thread.

4. (withdrawn) The method of claim 3, wherein the state information includes a pointer into the memory indicating where to move the first and second incoming blocks of data.

5. (withdrawn) The method of claim 4 further comprising:

storing data to memory in a sequential ordering based on the state information.

6. (withdrawn) The method of claim 5 further comprising:

providing the state information to transmit circuitry.

7. (previously amended) A method of processing a network packet received over a network at a processor having multiple programmable multi-threaded engines integrated within the processor, the method comprising:

processing a first portion of the network packet received at a port of a media access control device using a first thread provided by any of the multiple programmable multi-threaded engines integrated within the processor to move the first portion of the network packet to a first location in memory coupled to the processor; and

simultaneously processing a second portion of the network packet using a second thread provided by any of the multiple programmable multi-threaded engines integrated within the processor to move the second portion of the network packet to a second location in the memory.

8. (original) The method of claim 7 wherein the first thread and the second thread do not time share processing with one another.

9. (previously presented) The method of claim 8 wherein the first thread and the second thread operate out of different ones of the multiple multi-threaded engines integrated within the processor.

10. (original) The method of claim 7 wherein the first thread and the second thread time share processing with one another.

11. (previously presented) The method of claim 10 wherein the first thread and the second thread operate out of a common one of the multiple multi-threaded engines integrated within the processor.

12. (original) The method of claim 7 further comprising:

simultaneously with processing the first portion and the second portion of the network packet, processing a third portion of the network packet using a third thread.

13. (currently amended) The method of claim 12 wherein the first thread, the second thread, and the third thread run ~~the same~~ identical code.

14. (original) The method of claim 13 wherein the first thread, the second thread, and the third thread do not time share processing with each other.

15. (currently amended) An article comprising a computer-readable medium which store computer-executable instructions for receiving data from a plurality of ports, the instructions causing a processor having multiple programmable multi-threaded engines integrated within the processor~~, the method~~ to:

process a first portion of a data packet using a first thread provided by any of the multiple programmable multi-threaded engines integrated within the processor to move the first portion of the data packet to a first location in a memory coupled to the processor; and

process a second portion of the data packet using a second thread provided by any of the multiple programmable multi-threaded engines integrated within the processor to move the second portion of the data packet to the processor, wherein there is no time sharing between the first thread and the second thread.

16. (original) The article of claim 15, the article further comprises instructions to:

save state information of the first thread; and

restore the state information by the second thread.

17. (original) The article of claim 16, the article further comprises instructions to:

provide state information to transmit circuitry when an end of packet is detected by a subsequent thread.

18. (withdrawn) The method of claim 1, wherein the network packet comprises an Ethernet packet.

19. (withdrawn) The method of claim 1, further comprising monitoring the port of the media access control device for received data.

20. (withdrawn) The method of claim 1, wherein the processing comprises:

parsing the header of the received network packet;

performing a lookup based on the parsing; and

enqueuing an entry in a transmit queue for the network packet based on the performed lookup.

21. (previously presented) The method of claim 7, wherein each of the multiple programmable multi-threaded engines comprises an arithmetic logic unit, a control store, and multiple program counters associated with multiple corresponding threads provided by the engine.

22. (new) The method claim 7, wherein the network packet comprises an Ethernet packet and the media access control device comprises an Ethernet media access control device.